Renesas USB MCU

USB Host Mass Storage Class Driver (HMSC) Using Firmware Integration Technology Modules

R01AN2237JJ0111
Rev.1.11
Sep 30, 2015

要旨

本資料は、USB Host Mass Storage Class Driver Firmware Integration Technology（以降、USB HMSC FIT モジュールと記述）を使用したサンプルファームウェアの説明資料です。以降、本サンプルファームウェアを HMSC と記述します。

実際のソフトウェア開発時には、必ず” USB Host Mass Storage Class Driver (HMSC) Firmware Integration Technology アプリケーションノート(Document No.R01AN2026JJ)”、” USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート”(Document No. R01AN2025JJ)、及び各マイコンのユーザーズマニュアル(ハードウェア編)と併用してご利用ください。

対象デバイス

RX63N/RX631 グループ

RX64M グループ

RX71M グループ

本プログラムは Renesas Starter Kit (RSK)を使って動作確認を行っています。

目次

1. はじめに

1.1 機能概要

HMSC は、USB マスストレージクラスの Bulk-Only Transport（BOT）プロトコルで構築されています。ファイルシステム、ストレージデバイスドライバと組み合わせることで BOT 対応の USB マスストレージデバイスと通信が可能です。

HMSC が提供する機能を以下に示します。

- USB マスストレージクラスの BOT に対応。
- USB マスストレージサブクラスの SFF-8070i に対応。
- 1 本のパイプを IN/OUT で共用。
- サンプルアプリケーションを提供。(詳細は「3. サンプルアプリケーション」を参照ください)

1.2 HMSC 構成要素

HMSC は以下の FIT モジュールとサンプルアプリケーションで構成されています。

Table 1-1　HMSC 構成要素

FIT モジュール名	フォルダ名	バージョン
RX Family Board Support Package Module Using Firmware Integration Technology	r_bsp	2.81
Renesas USB MCU USB Basic Host and Peripheral firmware Firmware Integration Techology	r_usb_basic	1.11
Renesas USB MCU USB Host Mass Storage Class Driver (HMSC) Firmware Integration Techology	r_usb_hmsc	1.11
RX Family DTC Module Firmware Integration Techology	r_dtc_rx	2.02
RX Family Open Source FAT File System [M3S-TFAT-Tiny] Module Firmware Integration Technology	r_tfat_rx	3.01
RX Family M3S-TFAT-Tiny Memory Driver Interface Module Firmware Integration Technology	r_tfat_driver_rx	1.01

各 FIT モジュールの詳細は、関連ドキュメントを参照してください。また、本サンプルファームウェアで使用している FIT モジュールの最新バージョンは下記のホームページよりダウンロードが可能です。

ルネサスエレクトロニクスホームページ http://japan.renesas.com/

1.3 動作確認済環境

HMSC の動作確認環境を以下に示します。

＜評価ボード＞

Renesas Starter Kit+ for RX63N（RSK+RX63N）：型名:R0K50563NC000
RX63N グループ ルネサスマイコン開発スタータキット　ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX64M（RSK+RX64M）：型名:R0K50564MC010BR
RX64M グループ ルネサスマイコン開発スタータキット　ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX71M（RSK+RX71M）：型名: R0K5RX71MC010BR
RX71M グループ ルネサスマイコン開発スタータキット　ルネサスエレクトロニクス製

＜開発環境＞

統合環境 e² studio ルネサスエレクトロニクス製
RX ファミリ用 C/C++コンパイラパッケージ　Ver2.03.00　ルネサスエレクトロニクス製
E1 エミュレータまたは E20 エミュレータ　ルネサスエレクトロニクス製

＜その他＞

エミュレータ用ホスト PC（Windows® 7、Windows® 8、Windows® 8.1）

MSC デバイス(USB メモリ)
USB ケーブル
ユーザケーブル（E1 エミュレータまたは E20 エミュレータに同梱）
エミュレータ用ケーブル（E1 エミュレータまたは E20 エミュレータに同梱）

1.4 セットアップ

1.4.1 ハードウェア

HMSC の動作環境例をFigure 1-1に示します。評価ボードのセットアップ、エミュレータなどの使用方法については各取扱説明書を参照ください。

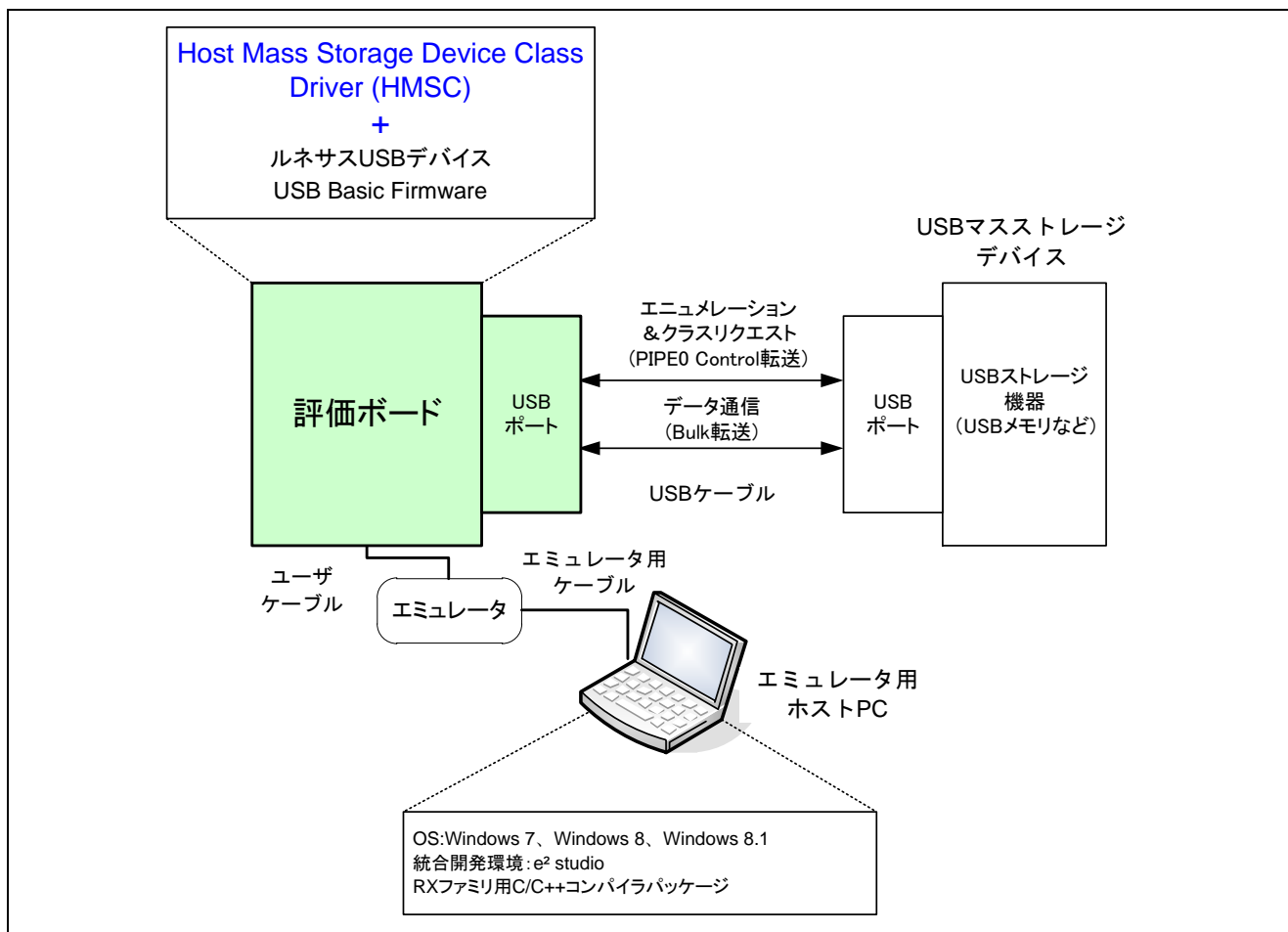



Figure 1-1 動作環境例

動作確認済みの評価ボードを**Table** 1-2に示します。

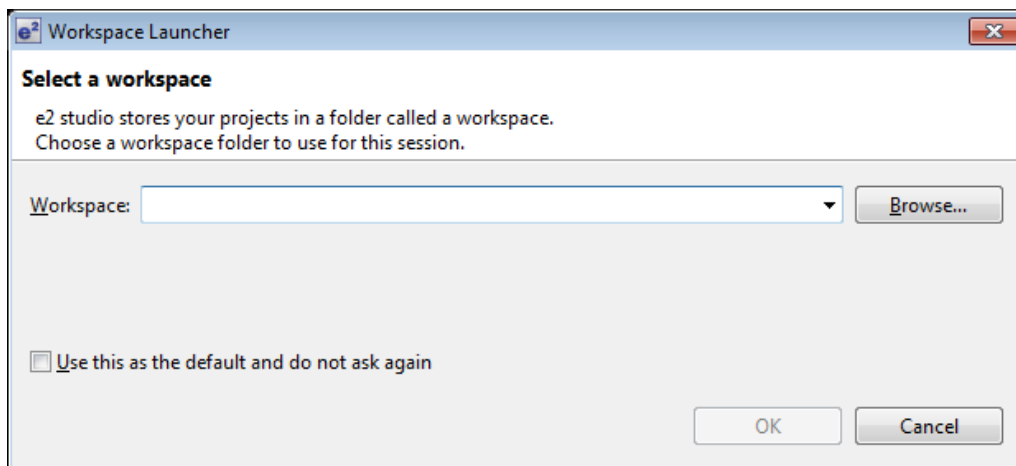
Table 1-2 HMSC 動作確認済みの評価ボード

マイコン	評価ボード
RX63N	RSK+RX63N
RX64M	RSK+RX64M
RX71M	RSK+RX71M

1.4.2 ソフトウェア

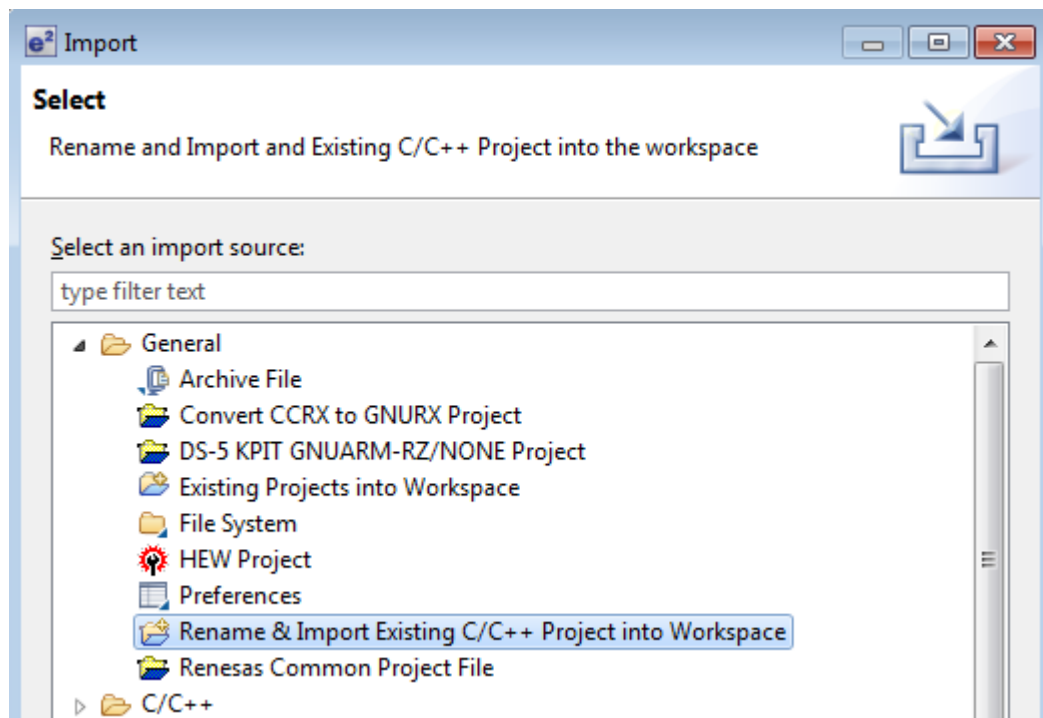
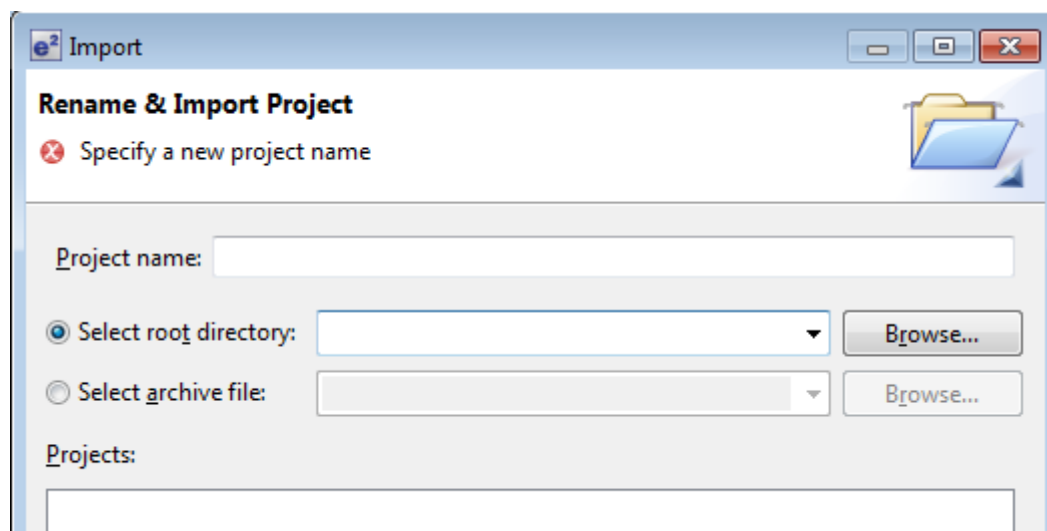
(1). e[2] studio を起動

a) e[2] studio を起動してください。

b) はじめて e[2] studio を起動する場合、Workspace Launcher ダイアログが表示されますので、プロジェクトを格納するためのフォルダを指定してください。

c) Workbench アイコンをクリックしてください。

(2). ワークスペースへのプロジェクトの登録

a) [File] --> [Import]を選択してください。

b) General => Rename & Import Existing C/C++ Project into Workspace を選択してください。

プロジェクトファイル".cproject"が格納されたフォルダを"Select root directory"に入力してください。

c) "Finish"をクリック

プロジェクトのワークスペースへのインポートが完了しました。同様の方法で他のプロジェクトを同一のワークスペースへインポートすることができます。

(3). "Build"ボタンをクリックし、実行プログラムを生成してください。

(4). デバッガへの接続を行い、実行プログラムをダウンロードしてください。"Run"ボタンをクリックすると、プログラムが実行されます。

2. ソフトウェア構成

2.1 モジュール構成

HDCD(ホストデバイスクラスドライバ)は HMSDD（ホストマスストレージデバイスドライバ）と HMSCD（USB ホストマスストレージクラスドライバ）の総称です。

Figure 2-1に HMSC のモジュール構成図、**Table 2-1**にモジュール機能概要を示します。

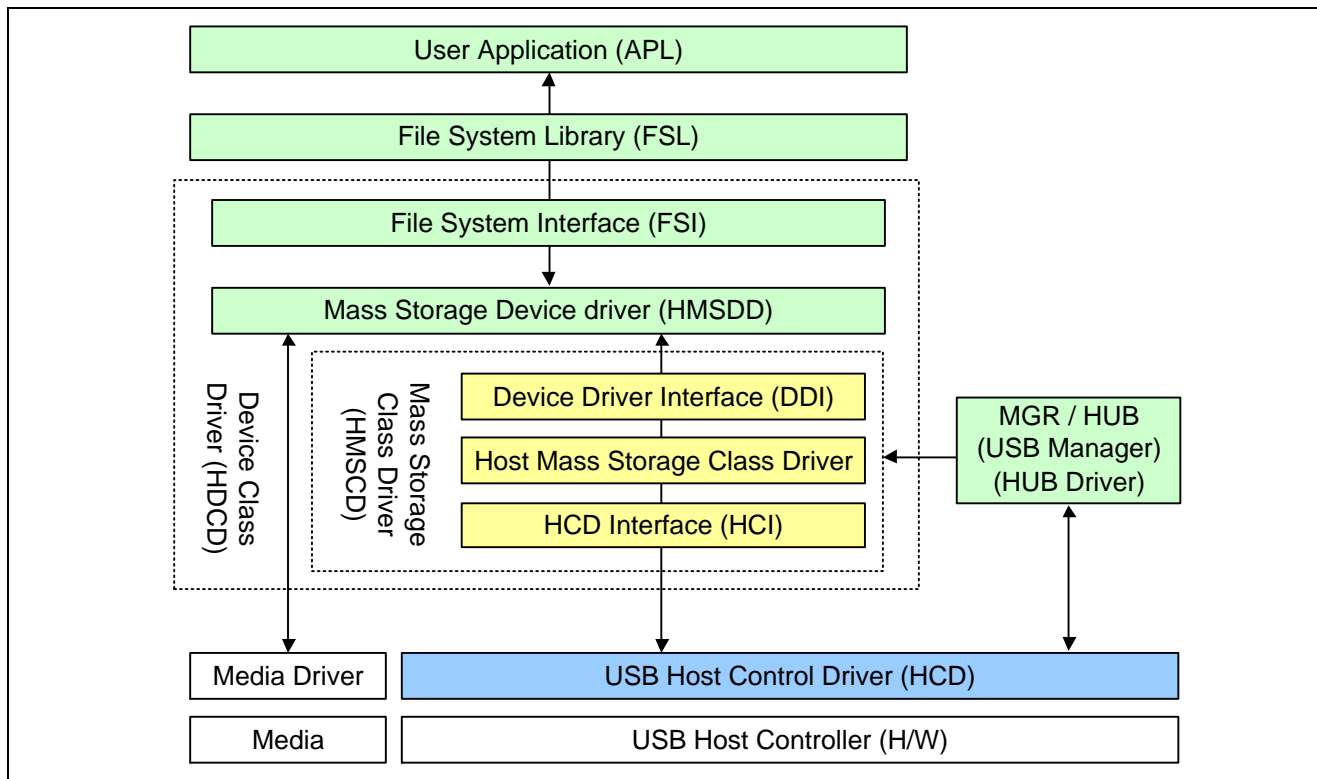



Figure 2-1 モジュール構成図

Table 2-1 モジュール機能概要

モジュール名	機能概要
APL	サンプルアプリケーションプログラム ・ FSL 関数を呼び出してストレージ機能を実現する
FSL	ユーザ仕様の FAT ファイルシステム
FSI	FSL-HMSDD 間のインターフェース関数
HMSDD	マスストレージデバイスドライバ
DDI	HMSDD-HMSCD 間のインターフェース関数
HMSCD	マスストレージクラスドライバ ・ ストレージコマンドに BOT プロトコルを付加して HCD に要求を行う ・ BOT のシーケンス管理を行う
HCI	HMSCD-HCD 間のインターフェース関数
HCD	USB Host H/W 制御ドライバ
Media Driver	USB でないストレージ機器ドライバ

3. サンプルアプリケーション

3.1 アプリケーション仕様

HMSC のサンプルアプリケーション(以降、APL)は、RSK 上に実装されたスイッチを使用し、スイッチ入力をトリガに動作する仕様になっています。
APL の主な機能を以下に示します。

1. MSC デバイスに対するエニュメレーション処理およびドライブ認識処理を行います。
2. スイッチ（SW2）押下によりファイル'hmscdemo.txt'を一回、MSC デバイスに対し書き込みます。
3. 上記書き込み後、ファイル'hmscdemo.txt'のリード処理を繰り返します。再度スイッチが押されるまでこのファイルリード処理を繰り返します。
4. ファイルリードを繰り返している状態で、スイッチ(SW2)を押下すると APL はファイルリードを停止します。
5. ファイルリードを停止している状態で、再度スイッチ(SW2)を押下すると APL はファイルリード処理を再開します。

[Note]

ファイル'hmscdemo.txt'が格納された MSC デバイスを接続し、上記の操作に従ってスイッチ(SW2)を押下した場合、'hmscdemo.txt'が上書きされます。

3.1.1 スイッチ

Table 3-1にスイッチ入力の仕様を示します。

Table 3-1 スイッチ入力による APL 動作内容

スイッチ名称	スイッチ番号	動作内容
ファイル操作スイッチ	スイッチ 2	スイッチが押下されると初回のみ Write 処理を 1 回実施し、その後 Read 処理を繰り返し行う

※評価ボードのスイッチと MCU ピン接続は、使用している RSK の取扱説明書、各 MCU のユーザーズマニュアルを参照してください。

3.1.2 MCU 端子設定

このアプリケーションプログラムでは、VBUS 端子および OVRCUR 端子の設定は r_usb_hmsc_apl.c ファイル内の usb_mcu_init 関数で行われています。

3.2 アプリケーション処理概要

APL は、初期設定、メインループの 2 つの部分から構成されます。以下にそれぞれの処理概要を示します。

3.2.1 初期設定

初期設定では、MCU の端子設定、USB ドライバの設定、USB コントローラの初期設定を行います。

3.2.2 メインループ

このメインループでは、以下の処理を行います。

1. APL は、ステートとそのステートに関連するイベントにより管理を行っています。APL では、まず、接続されたデバイスのステート (Table 3-2参照)の確認を行います。このステートは、APL が管理する構造体(“3.2.3 ステートとイベントの管理”参照)のメンバに格納されています。
2. 次に、APL は、そのステートに関連するイベント(Table 3-3参照)の確認を行い、そのイベントに応じた処理を行います。そのイベント処理後、APL は、必要に応じてステートを変化させます。このイベントは、APL が管理する構造体(“3.2.3 ステートとイベントの管理”参照)のメンバに格納されています。

以下に、APL の処理概要を示します。

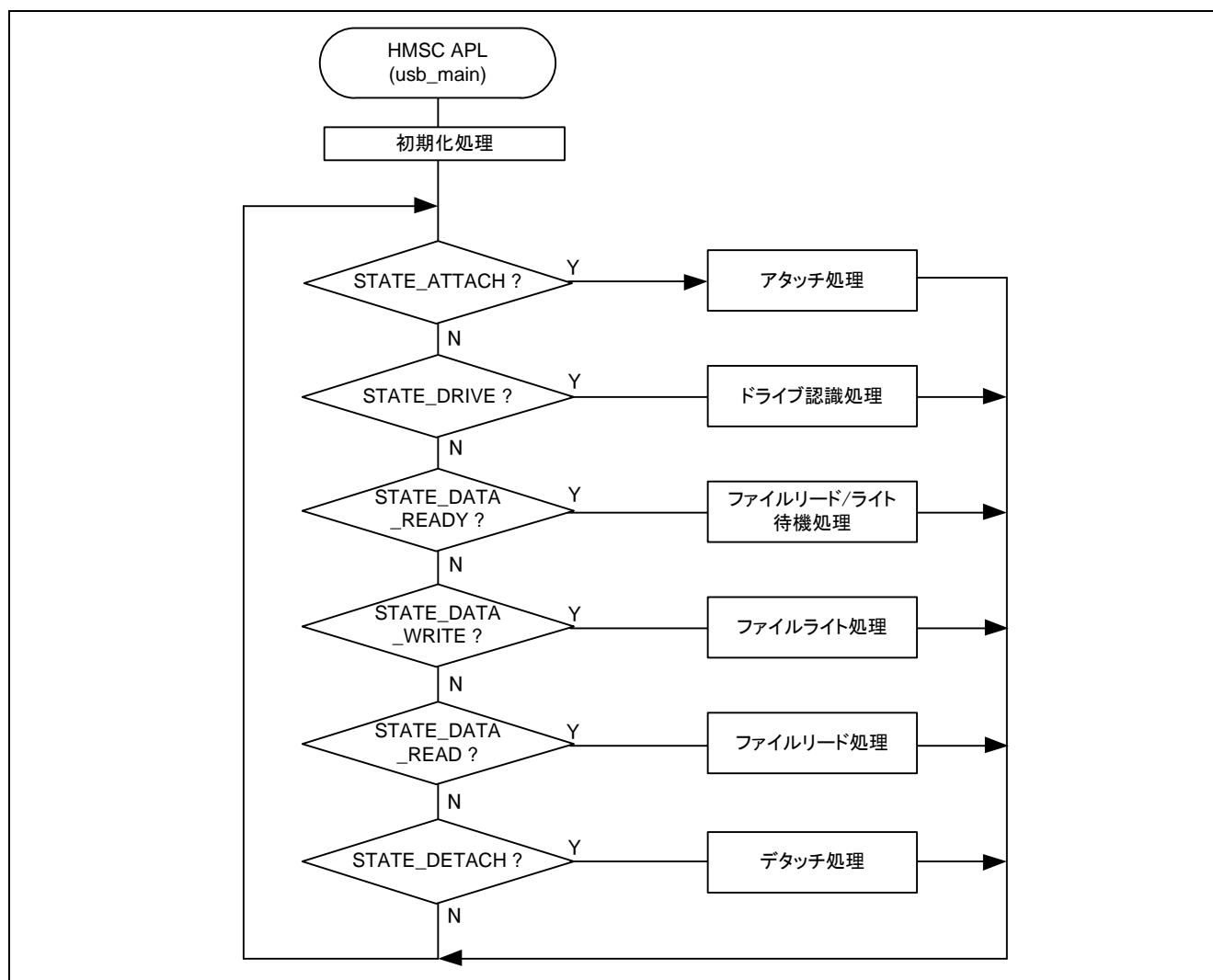



Figure 3-1 メインループ処理

Table 3-2 ステート一覧

ステート	ステート処理概要	関連イベント
STATE_ATTACH	アタッチ処理	EVENT_CONFIGURD
STATE_DRIVE	ドライブ認識	EVENT_DRIVE
		EVENT_DRIVE_COMPLETE
STATE_DATA_READ	ファイルリード	EVENT_READ_START
		EVENT_SWITCH_INPUT
STATE_DATA_WRITE	ファイルライト	EVENT_WRITE_START
STATE_DATA_READY	ファイルリード/ライト待機	EVENT_SW_INPUT
STATE_DETACH	デタッチ処理	--

Table 3-3 イベント一覧

イベント	概要
EVENT_CONFIGURD	USB デバイス接続完了
EVENT_DRIVE	ドライブ認識
EVENT_DRIVE_COMPLETE	ドライブ認識完了
EVENT_READ_START	ファイルリード
EVENT_WRITE_START	ファイルライト
EVENT_SWITCH_INPUT	スイッチ入力
EVENT_NONE	イベント無し

3.2.3 ステートとイベントの管理

ステートとイベントは、以下の構造体のメンバ(*state, event[]*)によって管理されています。この構造体は、APL が用意している構造体です。

```
typedef struct DEV_INFO                /* Structure for MSC device control */
{
    uint16_t    state;                 /* State for application */
    uint16_t    event_cnt;             /* Event count */
    uint16_t    event[EVENT_MAX];      /* Event. */
}
DEV_INFO_t;
```

[Note]

イベント取得処理で、取り出すイベントが存在しない場合、“EVENT_NONE”が返されます。

以下にステートごとの処理概要を示します。

1) アタッチ処理 (STATE_ATTACH)

== 概略 ==

このステートでは、RSK に MSC デバイスが ATTACH され、Enumeration が完了したことを APL に通知する処理を行い、ステートを STATE_DRIVE に変更します。

== 内容 ==

① APL では、はじめに初期化関数がステートを STATE_ATTACH にセットし、イベントを EVENT_NONE にセットします。
② MSC デバイスが接続されるまで STATE_ATTACH 状態が継続され、cdc_connect_wait()がコールされます。MSC デバイスが接続され、Enumeration が完了するとコールバック関数 msc_configured が USB ドライバよりコールされ、イベント EVENT_CONFIGURED を発行します。なお、このコールバック関数は、USB_HCDREG_t 構造体のメンバ devconfig に設定された関数です。
③ EVENT_CONFIGURD では、ステートを STATE_DRIVE に変更し、イベント EVENT_DRIVE を発行します。

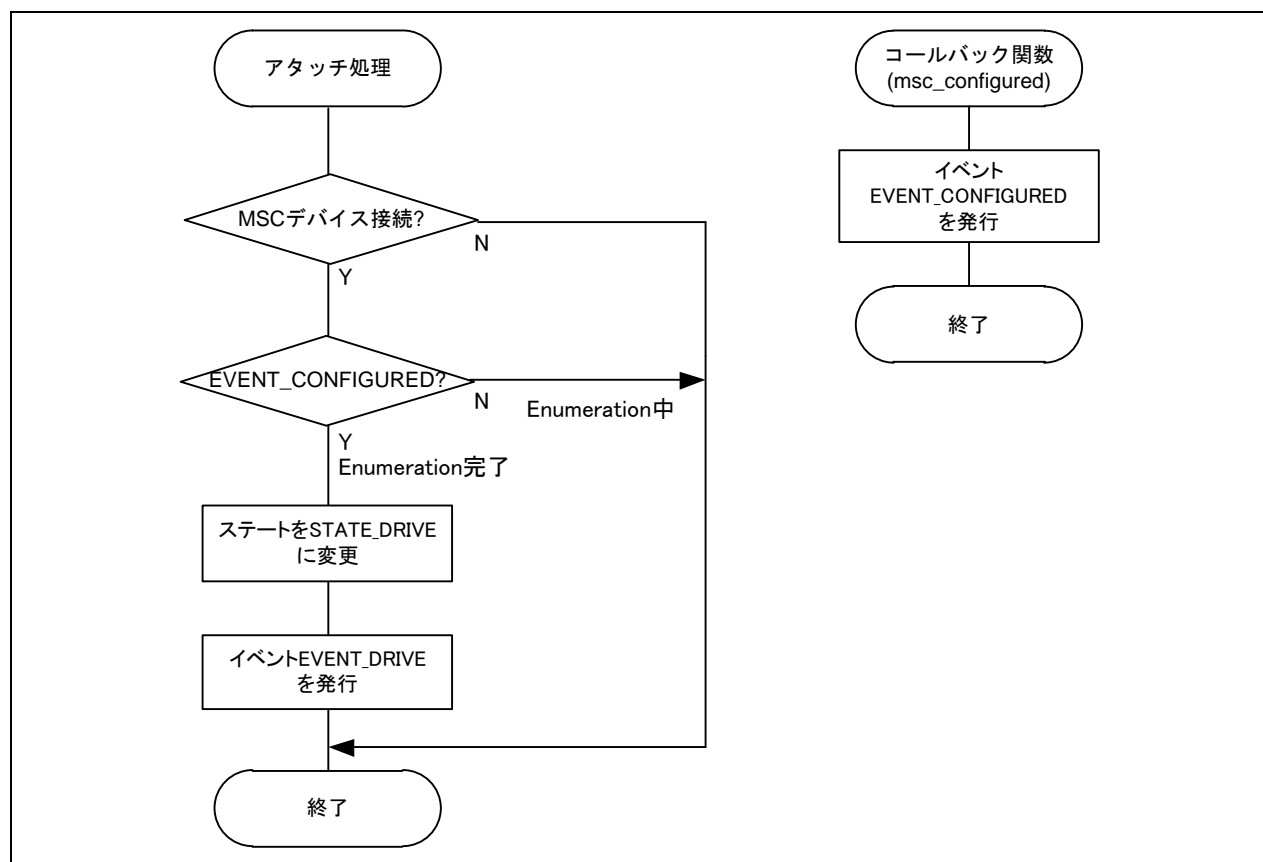



Figure 3-2 アタッチ処理概略フロー

2) ドライブ認識処理 (STATE_DRIVE)

== 概要 ==

このステートでは、MSC デバイスに対するクラスリクエスト GetMaxLUN の送信およびドライブ認識処理行い、ステートを STATE_DRIVE に変更します。

== 内容 ==

① このステートでは、はじめに、R_usb_hmsc_StrgDriveSearch()関数内で、クラスリクエスト Get Max LUN を MSC デバイスに送信します。クラスリクエスト Get Max LUN の送信が完了すると MSC デバイスに対し SFF-8070i のコマンドを MSC デバイスに送信し、ドライブ認識処理を行います。
② ドライブ認識処理が完了すると、コールバック関数 msc_class_request_complete がコールされ、EVENT_DRIVE_COMPLETE を発行します。
③ イベント EVENT_DRIVE_COMPLETE では R_tfat_f_mount()関数によるドライブマウント処理およびステート STATE_DATA_READY への変更を行います。

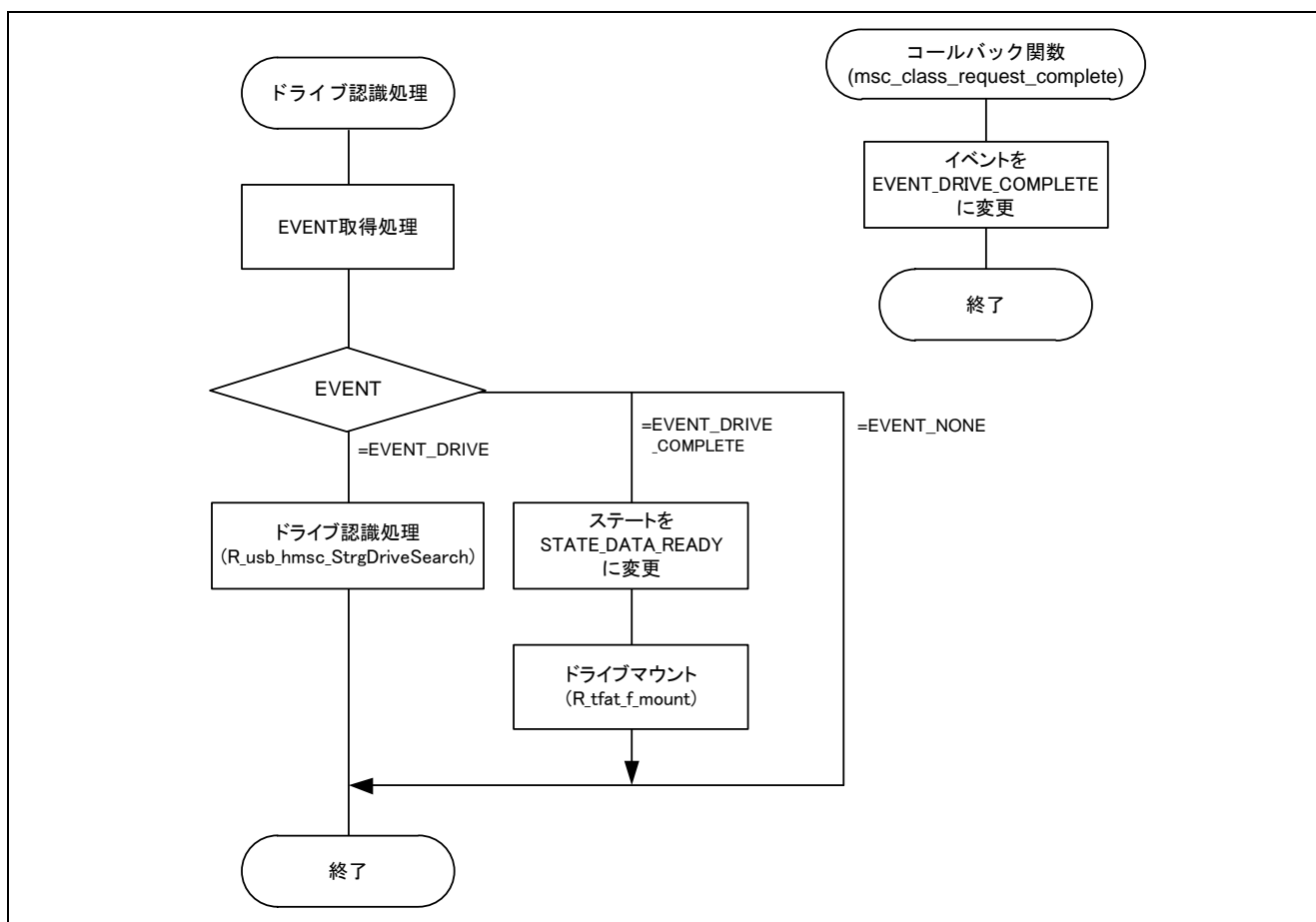



Figure 3-3 ドライブ認識概略フロー

3) ファイルリード/ライト待機処理 (STATE_DATA_READY)

== 概要 ==

このステートでは、STATE_DATA_WRITE に変更するか、STATE_DATA_READ に変更するかの判断処理を行います。

== 内容 ==

① スイッチ 2 が押下されることにより、EVENT_SWITCH_INPUT が処理されます。このイベントでは、ステートを STATE_DATA_WRITE に変更し、EVENT_WRITE_START を発行します。

② 接続した MSC デバイスに対しファイルライト処理を行った状態で、再度スイッチ 2 が押下されると、ステートを STATE_DATA_READ に変更し、EVENT_READ_START を発行します。

③ スイッチ 2 が押下されない場合、ステートは STATE_DATA_READY の状態を維持します。

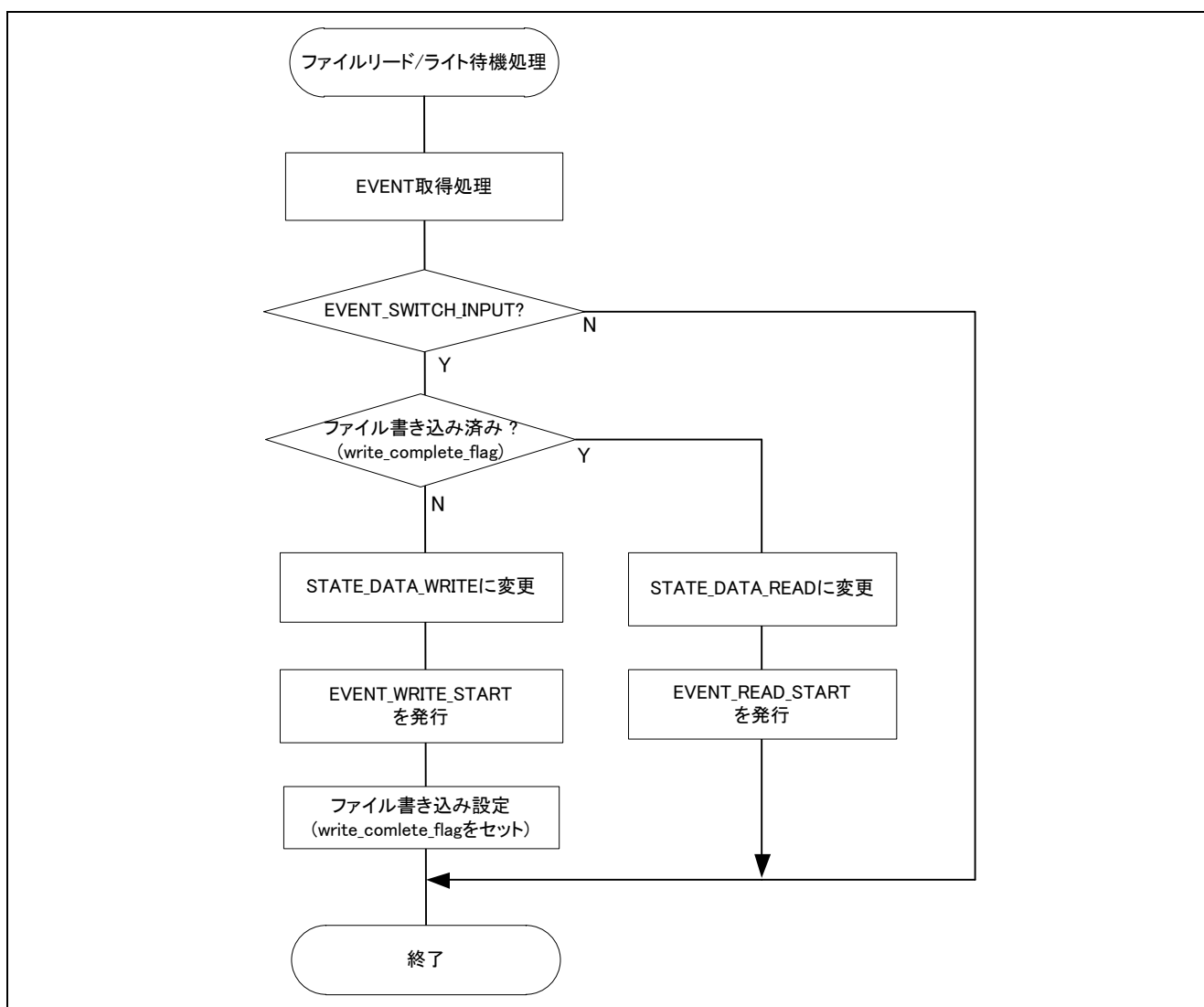



Figure 3-4 ファイルリード/ライト待機概略フロー

4)　ファイルライト処理 (STATE_DATA_WRITE)

== 概要 ==

このステートでは、接続した MSC デバイスに対しファイルライト処理を行い、ファイルライト完了後、ステートを STATE_DATA_READ に変更します。

== 内容 ==

① このステートでは、EVENT_WRITE_START が処理され、MSC デバイスに対してファイルライト処理を行います。
② ファイルライトが完了すると、ライトしたファイルをリードするため、ステートを STATE_DATA_READ に変更し、EVENT_READ_START を発行します。

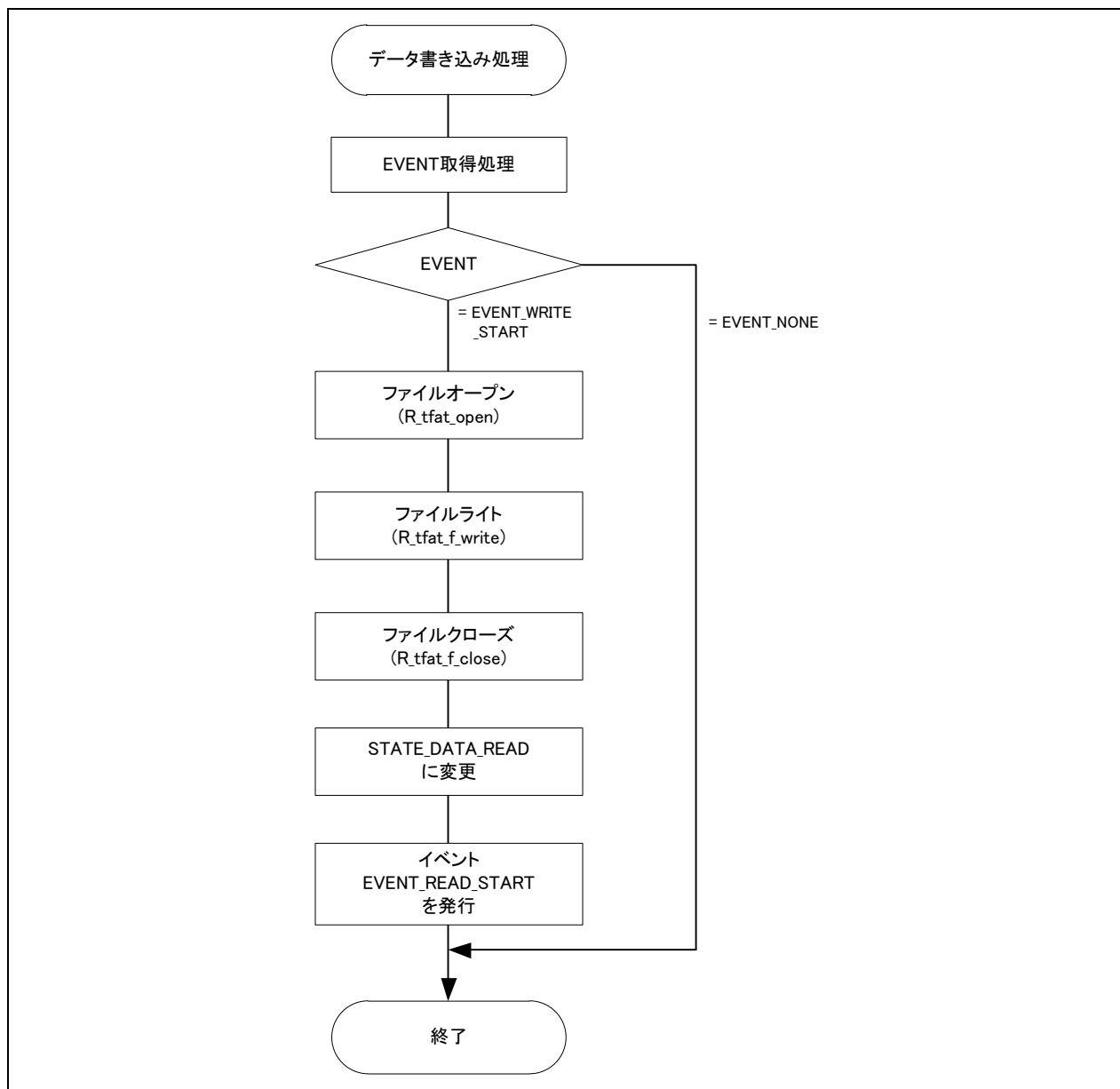



Figure 3-5　ファイルライト処理概略フロー

== 補足 ==

MSC デバイスに対するデータの書き込み/読み出しを行う場合、TFAT の API を使用しています。

5)　ファイルリード処理 (STATE_DATA_READ)

== 概要 ==

このステートでは、接続した MSC デバイスに対しファイルリード処理を行います。ファイルリード処理は、スイッチ 2 が押下されるまで継続されます。

== 内容 ==

① このステートでは、はじめに、EVENT_READ_START が処理され、MSC デバイスに対して書き込んだファイルのリード処理を行います。
② ファイルリードが完了すると、スイッチ 2 が押下されているかどうかを確認します。
③ スイッチ 2 が押下されていれば、EVENT_SWITCH_INPUT が処理され、ステートを STATE_DATA_READY に変更し、ファイルリード処理を停止します。スイッチ 2 が押下されていなければ、再度 EVENT_READ_START を発行し、ファイルリード処理を継続します。

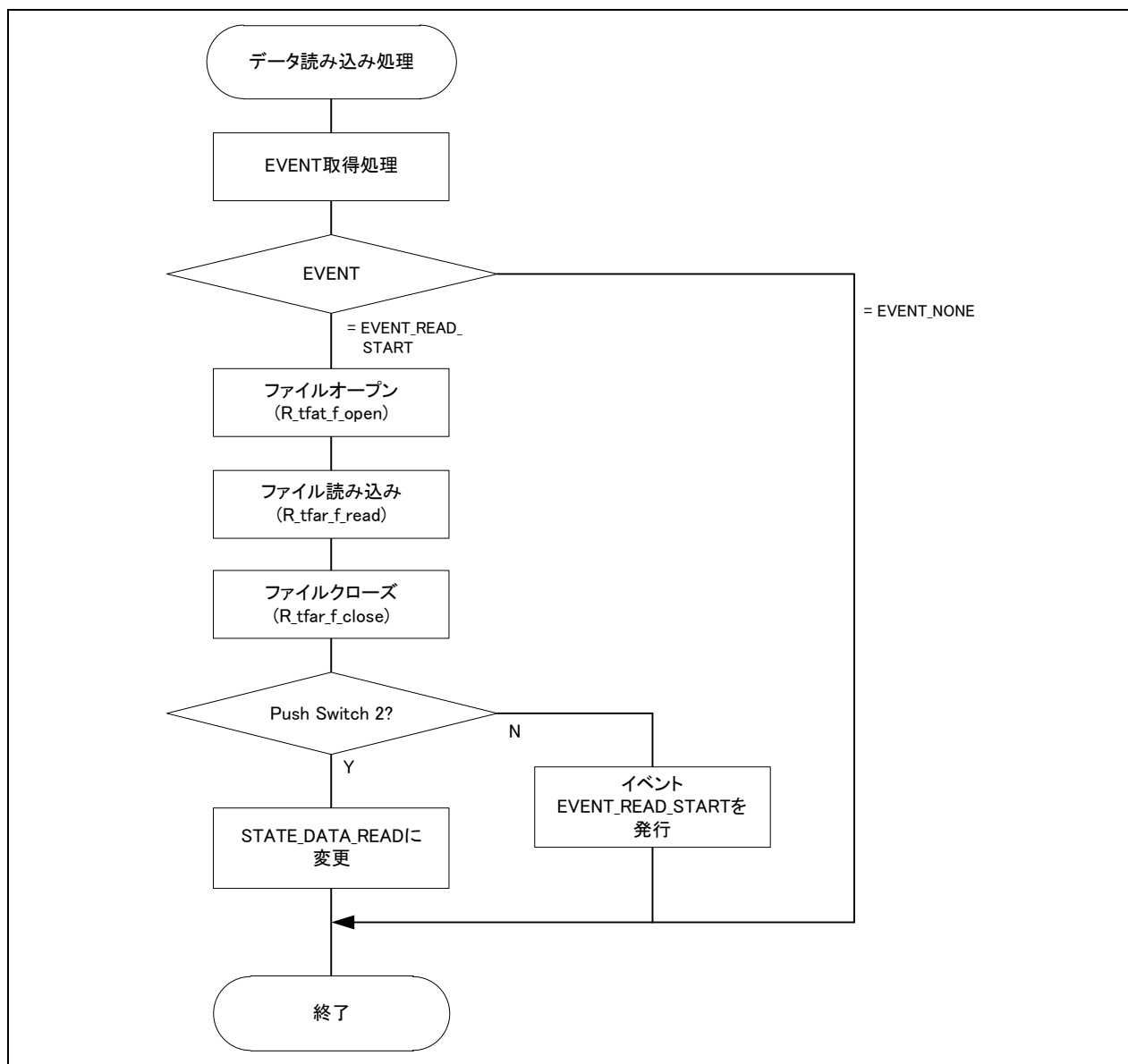



Figure 3-6　ファイルリード処理概略フロー

== 補足 ==

MSC デバイスに対するデータの書き込み/読み出しを行う場合、TFAT の API を使用しています。

6)　デタッチ処理 (STATE_DETACH)

接続された MSC デバイスがデタッチされると USB ドライバよりコールバック関数 msc_detach がコールされ、ステートを STATE_DETACH に変更します。STATE_DETACH では、特定の変数(msc_dev_info)にクリア処理、ステートを STATE_ATTACH に変更するなどの処理を行います。なお、コールバック関数 msc_detach は、USB_HCDREG_t 構造体のメンバ devdetach に設定した関数です。

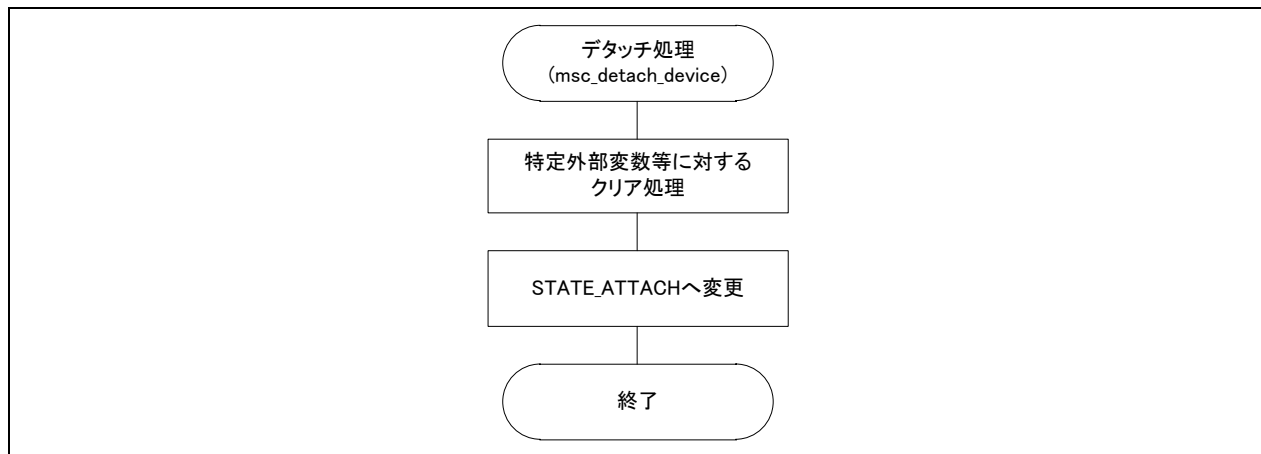



Figure 3-7　デタッチ処理概略フロー

3.3　注意事項

複数の MSC デバイスを接続する場合は、コンパイルするファイルを以下にとおりに変更してください。

r_usb_hmsc_apl.c　→　r_usb_hmsc_apl_mult.c

r_usb_hmsc_defep.c　→　r_usb_hmsc_defep_mult.c

4. クラスドライバ概要

4.1 クラスリクエスト

Table 4-1に HMSC が対応しているクラスリクエストを示します。

Table 4-1 クラスリクエスト

リクエスト	説明
GetMaxLun	サポートする最大ユニット数を取得する。
MassStrageReset	プロトコルエラーを解除する。

4.2 ストレージコマンド

HMSC は USB マスストレージデバイスへアクセスするために必須となるストレージコマンドと、サンプルのストレージコマンドに対応しています。
HMSC は USB マスストレージクラスの SFF-8070i に対応しています。

Table 4-2 サポートするストレージコマンド

コマンド名	コード	説明	対応（※）
TEST_UNIT_READY	0x00	ペリフェラル機器の状態確認	○
REQUEST_SENSE	0x03	ペリフェラル機器の状態取得	○
FORMAT_UNIT	0x04	論理ユニットのフォーマット	×
INQUIRY	0x12	論理ユニットのパラメータ情報取得	○
MODE_SELECT6	0x15	パラメータ指定	○
MODE_SENSE6	0x1A	論理ユニットのパラメータ取得	×
START_STOP_UNIT	0x1B	論理ユニットのアクセス許可／禁止	×
PREVENT_ALLOW	0x1E	メディアの取り出し許可／禁止	○
READ_FORMAT_CAPACITY	0x23	フォーマット可能な容量取得	○
READ_CAPACITY	0x25	論理ユニットの容量情報取得	○
READ10	0x28	データ読み出し	○
WRITE10	0x2A	データ書き込み	○
SEEK	0x2B	論理ブロックアドレスに移動	×
WRITE_AND_VERIFY	0x2E	確認付きデータ書き込み「	×
VERIFY10	0x2F	データ確認	×
MODE_SELECT10	0x55	パラメータ指定	×
MODE_SENSE10	0x5A	論理ユニットのパラメータ取得	○

※ ○:実装 ×:未実装

5. e² studio 用プロジェクトを CS+で使用する場合

HMSC のプロジェクトは、統合環境 e²studio で作成されています。HMSC を CS+で動作させる場合は、下記の手順にて読み込んでください。

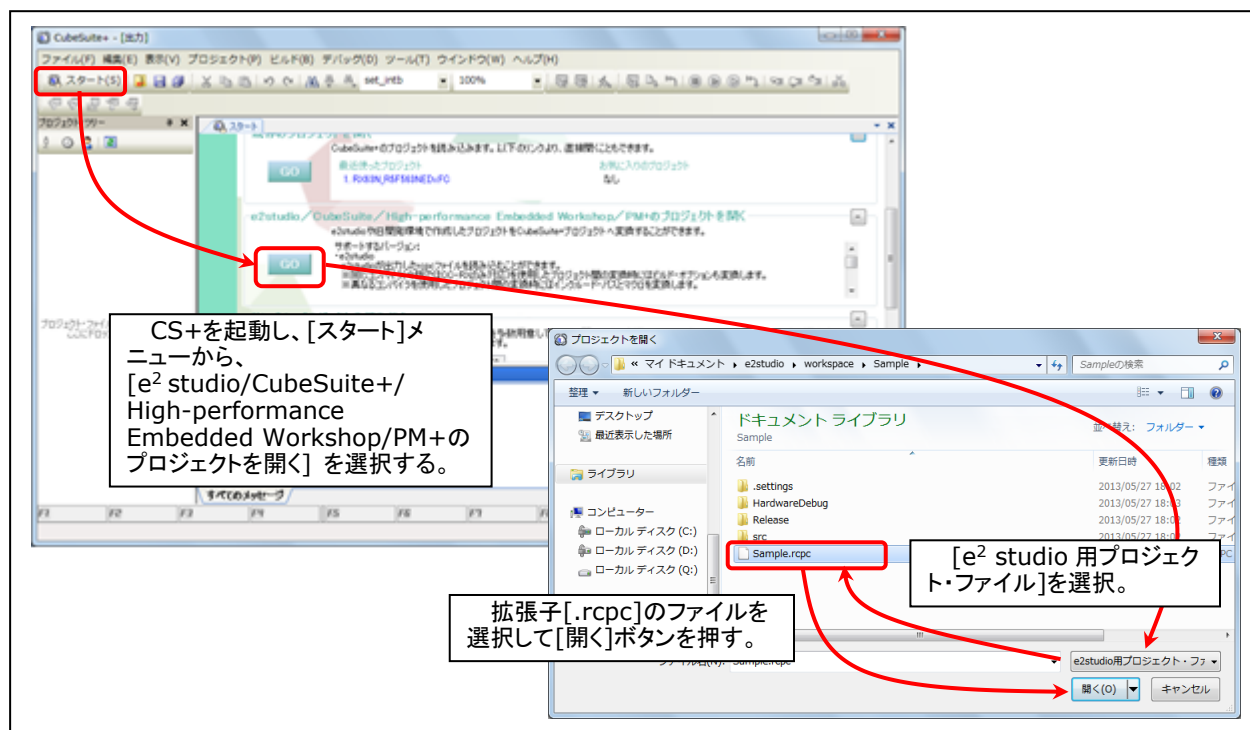



Figure 5-1 e² studio 用プロジェクトの CS+読み込み方法

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

http://japan.renesas.com/

お問合せ先

http://japan.renesas.com/contact/

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Oct 16, 2014	—	初版発行
1.10	Dec 26, 2014	—	RX71M を対象デバイスに追加。
1.11	Sep 30, 2015	—	RX63N/RX631 を対象デバイスに追加。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

   【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

   CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

   【注意】電源投入時は，製品の状態は不定です。

   電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

   外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

   同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

   【注意】リザーブアドレスのアクセスを禁止します。

   アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

   【注意】リセット時は、クロックが安定した後、リセットを解除してください。

   プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

   リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

   【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

   同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
   各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA機器、通信機器、計測機器、AV機器、
   家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。 たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。 なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
注2. 本資料において使用されている「当社製品」とは、注１において定義された当社の開発、製造製品をいいます。

ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/contact/

© 2015 Renesas Electronics Corporation. All rights reserved.
Colophon 4.0